Requirements Engineering 5: Requirements Elicitation

Steve Easterbrook

9/24/97

Outline

Importance of domain expertise

Importance of links with the customer(s)

Survey of elicitation techniques

Requirements Elicitation

- The requirements engineer must:
  - elicit knowledge about some problem domain
  - sufficient to analyze requirements for validity, consistency, completeness, etc.
  - I.e. become an expert in that domain
- Problems:
  - The knowledge is not always readily available
  - The knowledge might be distributed across many sources
  - It can be hard to get the knowledge from human experts
  - humans always introduce bias
  - There may be conflicts between knowledge from different sources

Source: *Adapted from Loucopoulos & Karakostas, 1995, p41*

Example

"The system shall accept *radar messages* from a *short-range radar*. The *scan-period* of the radar is 4 seconds. The *frequency* is 2.6-2.7 Ghz. The *pulse-repetition interval frequency* is 1040Hz. The *number of tracks* shall be for 200 aircraft. The *band-rate* is 2400. The *message size* is 104 bits/message. The system shall begin tracking aircraft that are within 2 miles of the controlled area. *Track initiation* will occur after 6 seconds."

Source: *Adapted from Loucopoulos & Karakostas, 1995, p40*

Customer-developer links

Custom software

- **Facilitated Workshop**
- **MIS intermediary**

Both

- **Support line**
- **Survey**
- **User interface prototyping**
- **Requirements prototyping**
- **Interview**
- **Testing**
- **Email/bulletin board**
- **Usability lab**
- **Observational study**

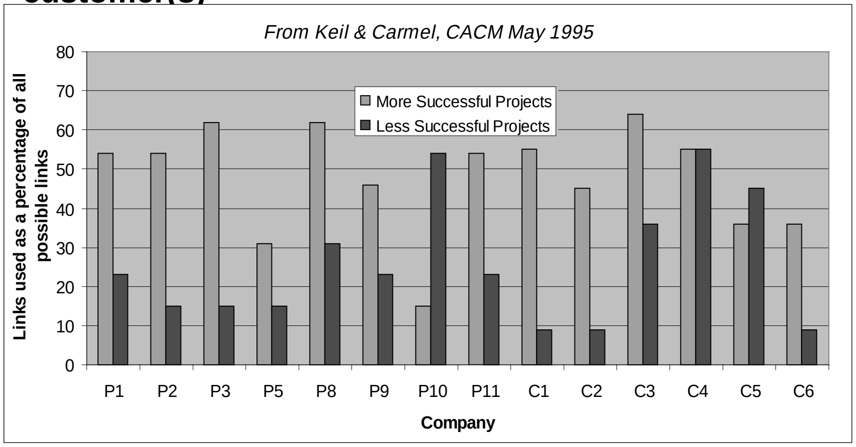
Package software

- **Marketing and sales meetings**
- **User group**
- **Trade Show**
- **Focus group**

Source: *Adapted from Keil and Carmel, 1995, p35*

Importance of links with customer(s)

- **Successful projects tend to have more links with customer(s)**



Source: *Adapted from Keil and Carmel, 1995, p37*

Elicitation Techniques

- Traditional Approaches
  - ➢ Introspection
  - ➢ Interview/survey
  - ➢ Group elicitation
- Observational approaches
  - ➢ Protocol analysis
  - ➢ Participant Observation (ethnomethodolog)
- Representation-based approaches
  - ➢ Goal-based
  - ➢ Scenario Based
  - ➢ Use Cases
- Approaches from Knowledge Engineering

Traditional Approaches

- Introspection
  - ➢ Very common, typical starting point for RE
  - ➢ Very poor at revealing what real users will need (eg Jirotka’s study)
- Interview/Survey
  - ➢ Questionnaires
  - ➢ Open-ended interview
  - ➢ Structured interviewing
- Group elicitation
  - ➢ Focus groups
  - ➢ Joint Application Development (JAD)
  - ➢ Brainstorming collective decision-making approach (BCDA)

Questionnaires

- Advantages
  - ➢ Can quickly collect info from large numbers of people
  - ➢ Can be administered remotely
- Disadvantages
  - ➢ Simplistic (presupposed) categories provide very little context
  - ➢ No room for users to convey their real needs
- Watch for:
  - ➢ Bias in sample selection
  - ➢ Bias in self-selecting respondents
  - ➢ Small sample size (lack of statistical significance)
  - ➢ Leading questions (“have you stopped beating your wife?”)
  - ➢ Appropriation (“What is this a picture of?”)
  - ➢ Ambiguous questions (I.e. not everyone is answering the same question)
  - ➢ *Questionnaires MUST be prototyped and tested*

Source: *Adapted from Goguen and Linde, 1993, p154.*

Interviews

- ➢ Structured - agenda of fairly open questions
- ➢ Open-ended - no pre-set agenda

- Advantages
  - ➢ Rich collection of information
- Disadvantages
  - ➢ Large amount of qualitative data can be hard to analyze
  - ➢ Hard to compare different respondents
- Watch for
  - ➢ Unanswerable questions (“how do you tie your shoelaces?”)
  - ➢ Tacit knowledge (“post-hoc rationalization”)
  - ➢ Removal from context
  - ➢ Interviewer’s attitude may cause bias

Source: *Adapted from Goguen and Linde, 1993, p154.*

Focus Groups, JAD, etc

- **Advantages**
  - ➢ **More natural interaction between people than formal interview**
  - ➢ **Can gauge reaction to stimulus materials (e.g. mock-ups, storyboards, etc)**
- **Disadvantages**
  - ➢ **Unnatural groups - may be uncomfortable**
  - ➢ **Groupthink**
  - ➢ **May only provide superficial responses to technical questions**
- **Watch for**
  - ➢ **sample bias**
  - ➢ **dominance and submission**
  - ➢ **consider using a trained facilitator**

Protocol Analysis

➢ **Think aloud protocols**
➢ **Retrospective protocols**

- **Advantages**
  - ➢ **direct verbalization of cognitive activities**
  - ➢ **Embedded in the work context**
  - ➢ **Very good at revealing interaction problems with existing systems**
- **Disadvantages**
  - ➢ **Introspection is notoriously unreliable**
  - ➢ **No social dimension**
- **Watch for**
  - ➢ **Poor (unrepresentative) choice of tasks**
  - ➢ **Observer bias (tendency to discount problematic phenomena)**
  - ➢ **Consider videoing for later playback and analysis**

Source: *Adapted from Goguen and Linde, 1993, p156.*

Ethnomethodology

- Basis
  - ➢ Social world is ordered
  - ➢ The social order may not be immediately obvious, nor describable from common sense
  - ➢ The social order cannot be assumed to have an a priori structure
  - ➢ I.e. social order emerges only when an observer immerses herself in it.
  - ➢ Emphasizes the importance of natural setting
- Categories
  - ➢ Most conventional approaches assume preexisting categories
  - ➢ This may mislead the observer (appropriation)
  - ➢ Ethnography attempts to use the subjects' own categories
  - ➢ Related to postmodern deconstruction: "there is no grand narrative"
- Measurement
  - ➢ No scientific objectivity, so use the subjects' own measurement theory

Source: *Adapted from Goguen and Linde, 1993, p158.*

Participant Observation

- Approach
  - ➢ Observer spends time with the subjects, joining in, long enough to become a member of the group ('longitudinal studies')
- Advantages
  - ➢ Contextualized;
  - ➢ Reveals details that other methods cannot
- Disadvantages
  - ➢ Extremely time consuming!
  - ➢ Resulting 'rich picture' is hard to analyze
  - ➢ Cannot say much about the results of proposed changes
- Watch for
  - ➢ going native!

Representational Techniques

- **Use Cases**
- **Scenarios**
- **Task Models**
- **Goals**

Use Cases

- **What is a use case?**
  - ➢ **Each different way that an actor interacts with a system is a use case**
  - ➢ **A description of a set of possible scenarios, with a common purpose**
  - ➢ **All the use cases need to be enumerated (or the requirements will not be complete)**
  - ➢ **Typically written in natural language**
  - ➢ **No internal description of the system; just the interaction.**
- **Combining use cases**
  - ➢ **extends/uses**
- **Advantages & Disadvantages**
  - ➢ **detailed characterization of all possible interaction with the system**
  - ➢ **helps in drawing system boundary, and scoping the requirements**
  - ➢ **Use cases do not capture domain knowledge**
  - ➢ **Don’t confuse use cases with a precise specification!**

Source: *Adapted from Rumbaugh 1997, p123-124*

Using Use Cases

- **Draw boundary**
  - ➢ identify actors outside the system boundary that interact with the system
- **For each actor**
  - ➢ identify possible use cases
  - ➢ make up some concrete scenarios to illustrate each use case
  - ➢ group similar scenarios into a use case if they are variations on a theme
- **For each use case**
  - ➢ write it up
  - ➢ specify rules for choice and iterations
  - ➢ consider all exceptions
  - ➢ look for overlap & commonalities with other use cases

Use Case Template

Name:

Summary:

Actors:

Preconditions:

Description:

Exceptions:

Postconditions:

Source: *Adapted from Rumbaugh 1997, p125-6*

Scenarios

- **Scenarios**
  - ➢ Specific sequence of interaction between actor and system
  - ➢ Tend to be short (e.g between 3 and 7 steps)
  - ➢ May be positive (I.e. required behavior) or negative (I.e an undesirable interaction)
  - ➢ May be indicative or optative
- **Advantages**
  - ➢ Very natural: stakeholders tend to use them spontaneously
  - ➢ Short scenarios very good for quickly illustrating specific interactions
- **Disadvantages**
  - ➢ Lack of structure: need use cases or task models to provide higher level view

Source: *Adapted from Dardenne, 1993.*

Task Models & Scenarios

- **Task Models:**
  - ➢ **hierarchical collections of stereotypical activities**
  - ➢ **Subgoals are tasks (or possibly use-cases)**
  - ➢ **Subgoals may occur in sequence, in parallel, or as alternatives; they may occur periodically or in response to contingencies.**
- **Scenarios:**
  - ➢ **are paths through a task model, taking in a specific time-sequence of steps**
  - ➢ **can be used to organize requirements**
  - ➢ **Can include parallelism**
  - ➢ **But can only include one alternative at each choice point.**
- **Exceptions**
  - ➢ **are important - often business critical - variants on the use case.**
  - ➢ **Cannot be modeled as scenarios themselves, as they interact with many concrete executable scenarios.**

Source: *Adapted from a message posted by Ian Alexander on the Software Requirements Engineering mailing list.*

Goal-based Approaches

- **Approach**
  - ➢ **Focus on why systems are constructed**
  - ➢ **Express the ‘why’ as a set of stakeholder goals**
  - ➢ **Use goal refinement to arrive at specific requirements**
  - ➢ **Goal analysis - document, organize and classify goals**
  - ➢ **Goal evolution - refine, elaborate, and operationalize goals**
  - ➢ **End up with a hierarchy of goals, showing refinement and obstacle relationships between them**
- **Advantages**
  - ➢ **Reasonably intuitive**
  - ➢ **Explicit declaration of goals provides sound basis for conflict resolution**
- **Disadvantages**
  - ➢ **Hard to cope with evolution of goals**

Source: *Adapted from Anton, 1996.*

Using a goal-based approach

- Goals
  - high level objectives of the business or organisation
- Requirements
  - specify how a goal is to be accomplished by the new system
- Types
  - Achievement goals
  - Maintenance goals
  - Soft goals
- Obstacles & constraints
  - Obstacles are behaviors that prevent achievement of a given goal
  - Constraints are conditions on the achievement of goals
- Tips
  - Multiple sources yield better goals
  - Associate stakeholders with each goal (reveals viewpoints and conflict)
  - Use scenarios to explore how goals can be met
  - Explicit consideration of obstacles helps to elicit exceptions

Source: *Adapted from Anton, 1996.*

Knowledge Elicitation Techniques

- Repertory grids
  - based on personal construct theory
  - constructs are attributes that people use to make distinctions in the world
  - develop a matrix: domain objects x attributes
  - Elicit constructs by taking objects in pairs or triples, and asking subjects how they would distinguish between them
- Proximity Scaling Techniques
  - help to elicit mental models, where complex multivariate data is concerned
  - very good for eliciting tacit knowledge
  - Given a set of domain objects, derives a set of dimensions for classifying them
  - step 1: pairwise proximity assessment among domain elements
  - step 2: automated analysis to build multi-dimensional space to classify the objects

Source: *Adapted from Hudlicka, 1996.*

Next Week

Requirements Modeling & Analysis

Notations

Comparison of Methods;

References

- Loucopoulos, P. and Karakostas, V. “System Requirements Engineering”. McGraw Hill, 1995.
- Keil M. and Carmel, E. “Customer-developer links in software development”. Communications of the ACM, Volume 38, No. 5, May 1995, Pages 33-44.
- Goguen, J. A., and Linde, C. “Techniques for Requirements Elicitation”. Proceedings, IEEE International Symposium on Requirements Engineering, San Diego, CA, 4-6 January 1993, pp152-164. Reprinted in Thayer, R. H and Dorfman, M. (eds.) “Software Requirements Engineering, Second Edition”. IEEE Computer Society Press, 1997.
- Rumbaugh, J. “Getting Started: Using use cases to capture requirements”. Journal of Object Oriented Programming, Sept 1994. Reprinted in Thayer, R. H and Dorfman, M. (eds.) “Software Requirements Engineering, Second Edition”. IEEE Computer Society Press, 1997.
- Dardenne, A. “On the Use of Scenarios in Requirements Acquisition”. Technical Report #CIS-TR-93-17, Dept of Computer Science, University of Oregon, 1993.
- Anton, A. “Goal-based Requirements Analysis”. Proceedings, Second International Conference on Requirements Engineering, Colorado Springs, CO, April 15-18, 1996.
- Hudlicka, E. “Requirements Elicitation with Indirect Knowledge Elicitation Techniques: Comparison of Three Methods”. Proceedings, Second International Conference on Requirements Engineering, Colorado Springs, CO, April 15-18, 1996.